5by5 Networks

NGCC v1.0

Agent desk top – Functional Specifications

V 03

Contents
Agent Application Approach

Agent Desktop Application

Introduction

The Agent user interface will have the following distinct sections, which will drive a specific functionality.

1. My Presence Area
    a. Displays information on Configured channels v/s active channels
2. Presence Area
    a. Displays presence information of the team
    b. Displays real time statistics of the team and Call center
3. Reader Board
    a. Displays real time messages from the Personal reader board services
4. Applications area
    a. Enterprise applications accessed through browser
5. Media Control area
    a. Controls for Channel specific media control
6. Wrap up area
    a. Entry area for wrap up of the call

In addition to the UI driven functionality, the agent application need to implement the following services.

1. Log In services
    a. Ability to log in to the following systems
        i. RTC server
        ii. Sharepoint services server
        iii. Linked Enterprise applications
    b. Provide single sign on capabilities, if possible
2. Message interaction services
    a. .NET remote message interface for the UQE and the Call managers to communicate with Agent desktop.
    b. Use MSMQ message interface to publish the events to persist in the database
3. Application audit / management services
    a. Log the screen shot of the agent at a pre defined events / time interval.
    b. Log all the application errors in to Application events log in Ver 1.0 and in to a central repository in Ver 2.0
    c. Enable the agent desktop application to be monitored by the supervisor in real time. Ability for the supervisor to see the Agents application interaction, with out the knowledge of the Agent.
4. Application launch services
    a. Ability to launch another desk top application by passing appropriate parameters from the agent application. The pre configured parameters are
        i. #CUSTOMER_ID

ii. #AGENT_ID
iii. #CALL_CENTER_ID
iv. #ENTERPRISE_ID
v. #CALL_NO

The agent application will implement accelerated keys for specific functionality. Some of the keys are:

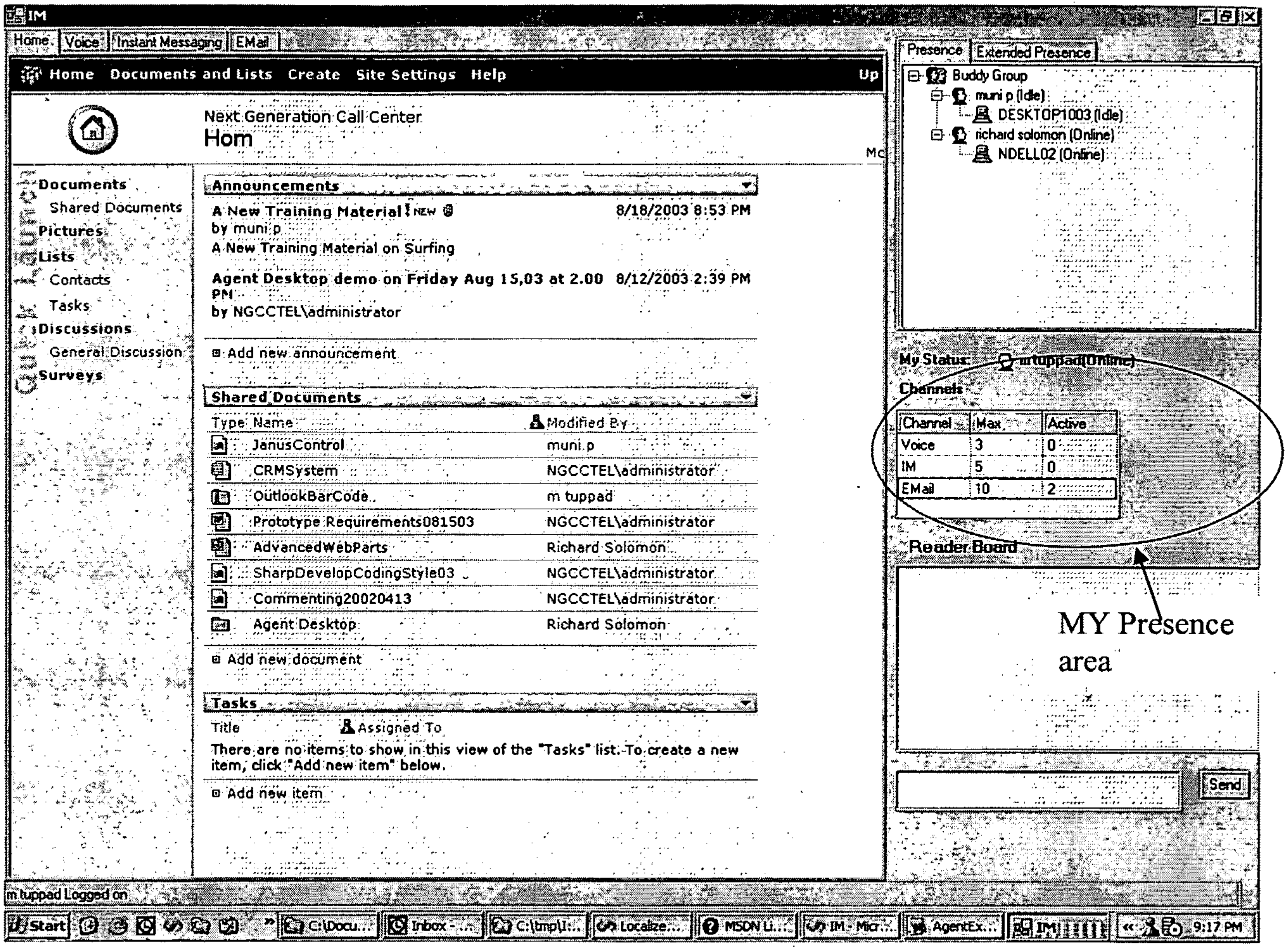
- <Alt> C – Close current call
- <Alt> A – Accept the Call
- <Alt> F – Forward the call
- <Alt> S – Consult supervisor
- <Alt>T – Request transfer

The agent application will be deployed through .NET application distribution capabilities, where by there will be a small assembly distributed to the agents. This assembly always, refers to the application DLL stored on the sharepoint services server. On invocation for the first time, the application will be downloaded to the client and run at client. During subsequent invocation of the application, if the application version is changed, then the new version is downloaded on to the agent desk top other wise, the existing application will be launched from the local application cache. With this, when ever there is update to the agent desk top application, we will have to load the new version on to the central share point services server, the distribution will take place automatically.

Detailed functionality

User Interface

The sample UI is as shown below:



Next Generation Call Center
Hom

Documents
Shared Documents
Pictures
Lists
Contacts
Tasks
Discussions
General Discussion
Surveys

Announcements

A New Training Material! NEW 8/18/2003 8:53 PM
by muni.p
A New Training Material on Surfing

Agent Desktop demo on Friday Aug 15,03 at 2.00 PM 8/12/2003 2:39 PM
by NGCCTEL\administrator

Add new announcement

Shared Documents

Type	Name	Modified By
	JanusControl	muni.p
	CRMSystem	NGCCTEL\administrator
	OutlookBarCode	m tuppad
	Prototype Requirements081503	NGCCTEL\administrator
	AdvancedWebParts	Richard Solomon
	SharpDevelopCodingStyle03	NGCCTEL\administrator
	Commenting20020413	NGCCTEL\administrator
	Agent Desktop	Richard Solomon

Add new document

Tasks

Title Assigned To

There are no items to show in this view of the "Tasks" list. To create a new item, click "Add new item" below.

Add new item

Buddy Group
muni p (Idle)
DESKTOP1003 (Idle)
richard solomon (Online)
NDELL02 (Online)

My Status: mtuppad(Online)

Channels

Channel	Max	Active
Voice	3	0
IM	5	0
EMail	10	2

Reader Board

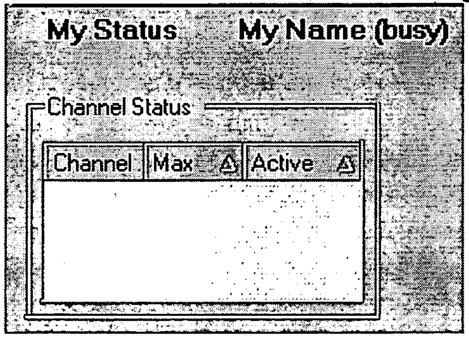
MY Presence area

m tuppad Logged on

My Presence area

This area contains two entries:

- The name of the logged in person and the overall status.
- Channel wise the total configured and the currently engaged



My Status My Name (busy)

Channel Status

Channel	Max	Active

Overall status

The overall status will be displayed as a dropdown list box. The agent will be able to select the overall status manually. This will help the agent to communicate the agent's presence changes due to external activities like Lunch, Meeting, Going to Rest room etc.

The application will also change the overall status based on several factors. It is calculated based on the configurations. In ver 1.0 we will support the following configurations:

- When there is one incoming call on Voice Channel
- When no call on voice but the Active IM sessions are equal to the maximum sessions set for the logged in individual.

Channel wise status

Will be shown as a table with one row per each channel. With attributes as Maximum allowed and the Active channels. The List of channel types enabled and the maximum allowed for each channel is based on the Agent channel capabilities and the workstation, from where the agent is logged in. The minimum capabilities of the two is taken as the maximum possibilities. This data will be taken from the data services during the agent log in process and set as part of the Agent's extended presence properties (Part of notes in RTC File). We will use the extended properties XML format for this purpose. Populate only the applicable nodes / elements in the xml.

The Active channels

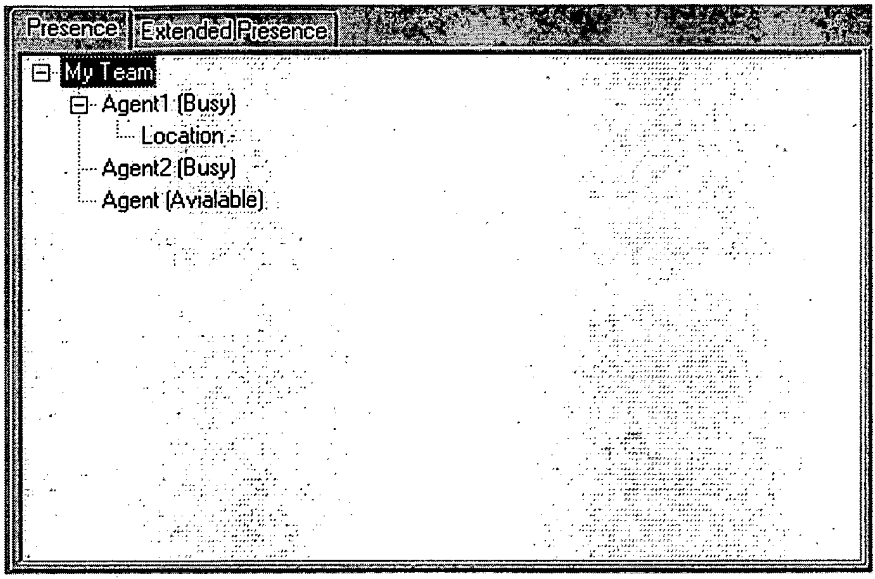
This data is calculated based on the actual interactions being taken place. For eg. If the voice call is routed to the agent desktop, then the There will be one entry in extended presence with voice call details and the Active under the voice channel is incremented by 1. When the call is closed, the call entry in the extended presence is removed, the number used under voice channel is decremented by 1.

Real Time Reporting area

Will have three tabs.

- BAM Focus
- BAM Details
- Person Search

BAM Focus tab



Shows the groups, on which the agent can see real time statistics, these groups are as set up by the supervisor. The groups may be call centers, agent groups. The list will be shown in a tree structure as shown. When double clicked on group or the individual person / name in the group, the control will be transferred to 'BAM Details' Tab.

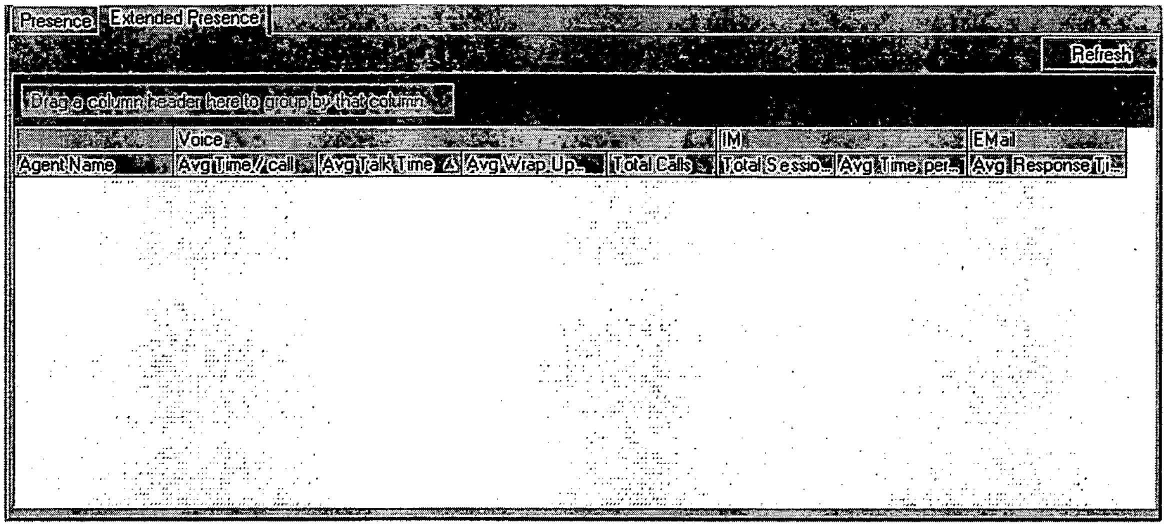
There may be several types of buddy lists,

- Agent Statistics
- Call center
- Enterprise Statistics.
- Call awaiting assignments
- My Buddies

The parameters applicable to each type are configured in the database. The real time values are stored as part of the extended presence of each of the objects

Under the My Buddies group the agent can add his/her own buddies by giving exact buddy ID.

BAM Details Tab



Drag a column header here to group by that column.

	Voice				IM		EMail
Agent Name	Avg Time / call	Avg Talk Time	Avg Wrap Up...	Total Calls	Total Sessio...	Avg Time per...	Avg Response Ti...

This tab, contains a table with the first column list of persons / names in the group, followed by series of columns with a back ground color based on real time statistics.

At log in the parameters for each type and the formatting rules for each is fetched, the respective table columns and the column format rules are populated for each type. There will be one default rule. If no value then background color is 'Grey'.
In version 1.0 we support four color scheme approaches.

- Green – where the actual parameter is with in the limits
- Yellow – Where the parameter is in border case
- Red – Where the parameter is out of bound.
- Grey – Where the value is not applicable.

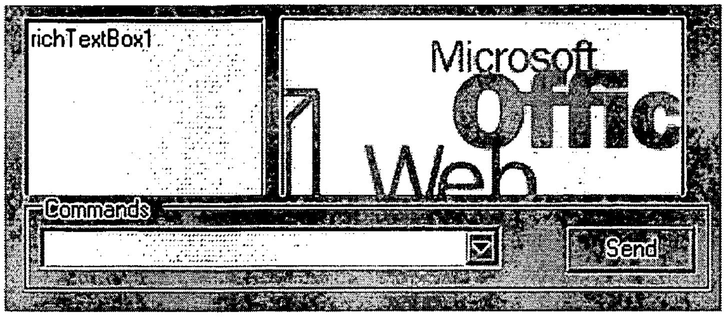
This color scheme is defined for each attribute at each contact center / enterprise level. These are reconfigurable by the site administrators.

When traversed to this tab, the names of buddies are fetched from the buddy list. The real time data is fetched from the extended presence for each of the member of the buddy list and the appropriate cells are populated with the actual data. The sample screen shot is as shown below.

Person Search Tab

Under this tab, we will provide the capability for the agent to search persons in the call center. The search criteria will be First Name, Last Name, ID. We will allow the starting with string search capabilities. The search results will display the profile of the person with details of First Name, Last Name, Phone and ID. If more than one person is found with the given criteria, then all the people are shown in the list. When double clicked on the name, the person is added to the Agent's My Buddies list.

Reader Board



Is the IM Message window established with a NGCC BOT service which provides Personal Reader board services. There will be two types of readerboard services.

- Schedule based services
- On Demand services

At the time of log in, a IM Session is established with the NGCC BOT, which will be running as another RTC User and that session is attached to this area.

Schedule based services

In Version 1.0 we will implement pre configured NGCC BOT services. Where the supervisor can configure this service for each of the agents or all the agents in the group. The configuration will include

- The message type, which may include a statistic or a group of pre configured statistics.
- First message start time, Validity period. The frequency of distribution in Minutes.

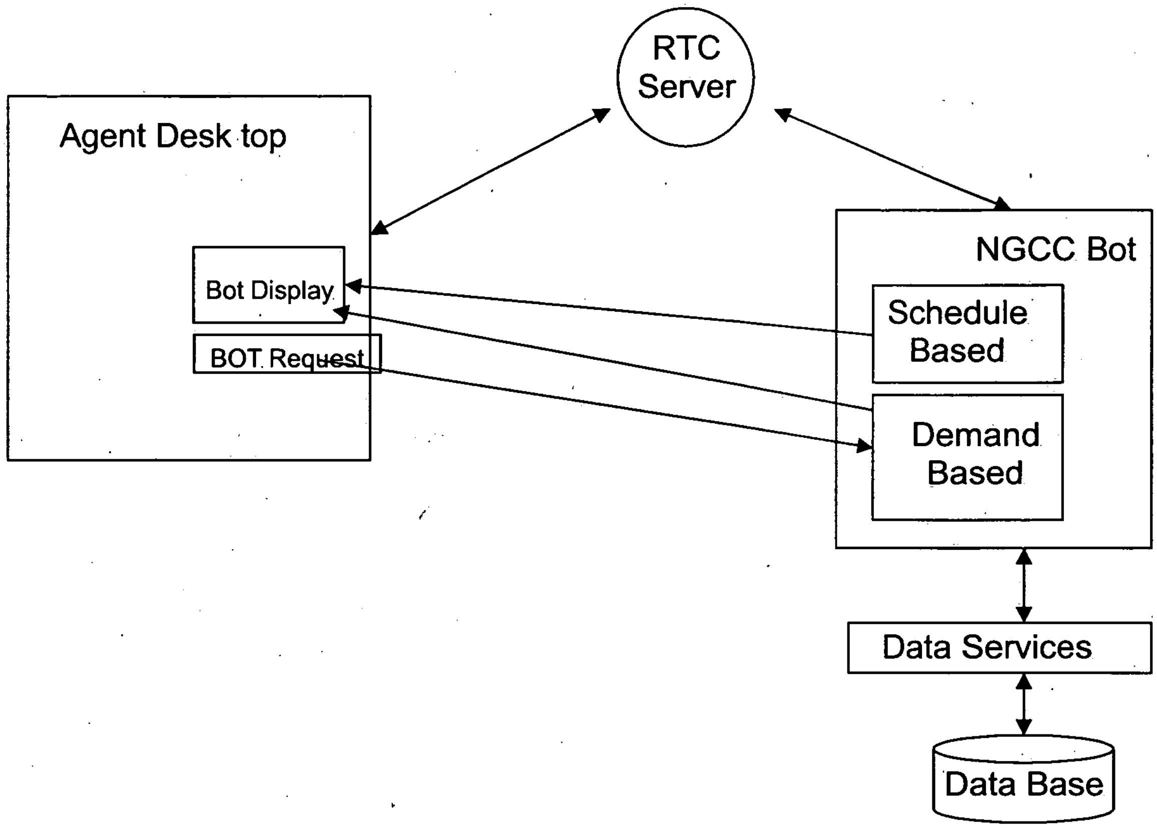
Based on this configuration, the NGCC BOT will keep sending the messages to the agents in a periodic basis. Agent desktop will display these messages in a appropriate graphical form, similar to extended presence display format.

On Demand Services

Here the agent will request the NGCC BOT to send a specific measure. Here we will provide a list of configured BOT services as drop down list box, stored in the database, loaded at the time of log in.
The following block diagram shows the various components involved and their relationship.
(We need to work out the technology implementation details for NGCC BOT Service).

Others

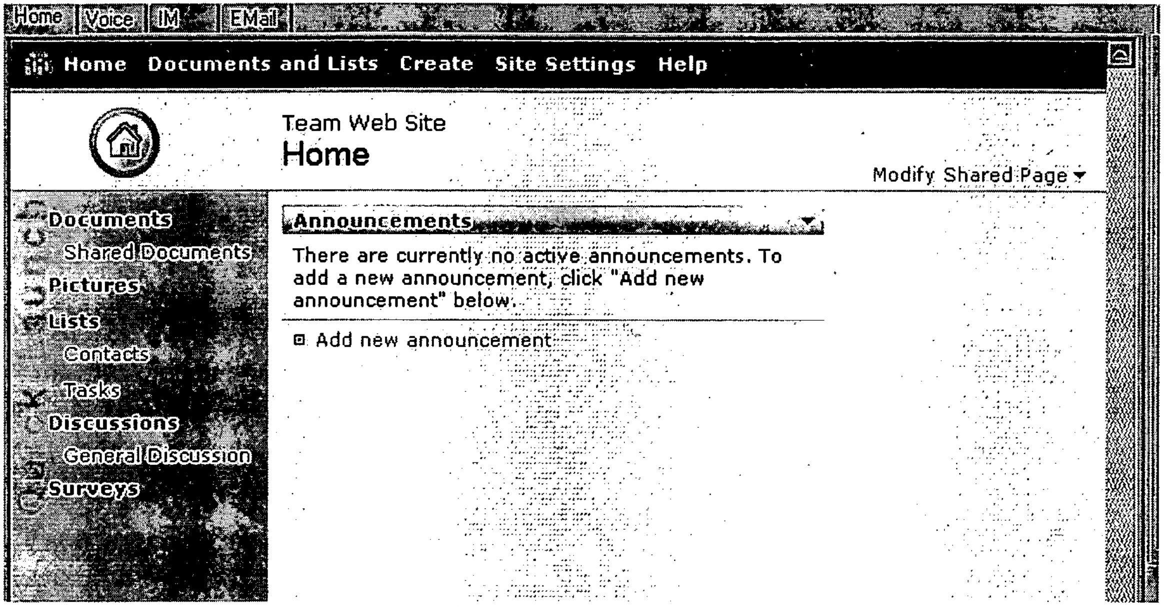
The reader board services may be extended to send Alerts and Notifications to the people on reaching a certain pre defined thresholds and also on encounter of certain type of errors.



Application Area

Application area represents the area where the agent desk top is linked with enterprise applications. There will be one many tabs based on the number of current sessions going on.

Home Tab



There are currently no active announcements. To add a new announcement, click "Add new announcement" below.

When the agent is logged in to the system or when agents does not have any on going sessions, the agents will be placed on this tab.
Is the tab which links to the Agent's home contact center web site. Which will be running on Share Point services. On this page we can configure many useful components for the CCO. Some of the useful components are.

- Announcements - The CCO Management / supervisor's can use this component to communicate to the agents on a regular basis, the content can also be set with a start time and end time, so that that content will appear only during that specific period. This can be effectively used to share company news, upcoming events, projects etc.
- Bulletin board

Channel Tabs

There will be several tabs, depending on the active sessions. Each tab will be directly related to a particular call / session. Each tab will have different layout, depending on the type of channel.

When a new request is routed to the agent by NGCC through .NET remote API call, the following possibilities may occur.
If the agent has set his/her presence manually and the overall status is not in 'Available' State then the call is rejected.
If the agent is available, but the agent is already full with the capacity for the incoming channel, the active sessions under each channel are equivalent to the maximum possible, then the Agent message interface will directly 'reject' the call. No action is taken.
If the capacity is available, then the following actions will take place:

- In the extended presence, a entry for that channel type is added with all the attributes, channel status is changed to 'Reserved'.
- Overall status is computed and the status is set.

- A new channel specific application area tab is activated with the appropriate parameters.
- The tab title is set with Channel + Contact ID. (Voice – 12)
- If there are no active sessions, then the focus is set to new tab, to bring the tab to front otherwise the tab title background color is changed to AGENT_NEW_CALL_BG

the active sessions are incremented and the message parameters are passed
Voice Channel tab format is as shown below.

IM Channel tab format

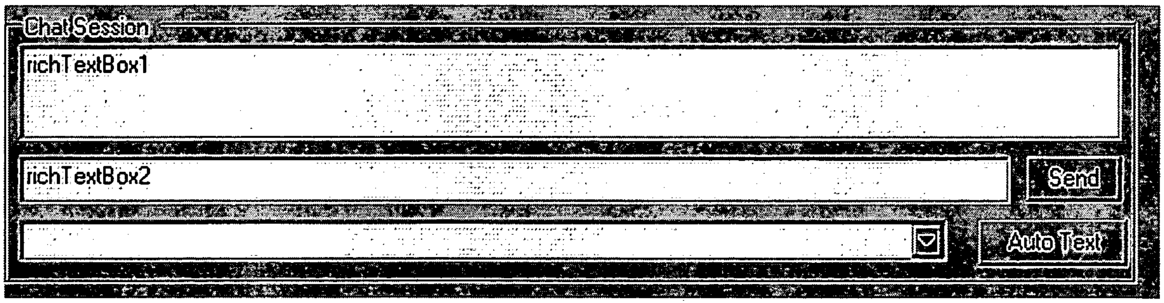
In Case of, the messages are asynchronous and there is a possibility of several sessions at the same time. To provide, a easy way for the agent to navigate through these sessions we will follow the following color coding for the background color of the Tab.

- When the agent receives a message from the customer, the background color will be set to IM_T*AB*_BG_RECD
- When the agent sends the message to the customer, the background color will be set to IM_T*AB*_BG_SEND

The transfer of control between the tabs will happen only with the agent clicking on the tab. When the new message is received, and if there are already ongoing messages, then the

The constants (IM_T*AB*_BG_SEND , AGENT_NEW_CALL_BG , IM_T*AB*_BG_RECD) will be defined in the configuration table and will be accessed as part of the log in process.

IM Message area



In the IM Channel tab, there will be a message area for the agent to communicate with the customer. In this area there will be one bib text box, where all the messages are stored in a chronological order.

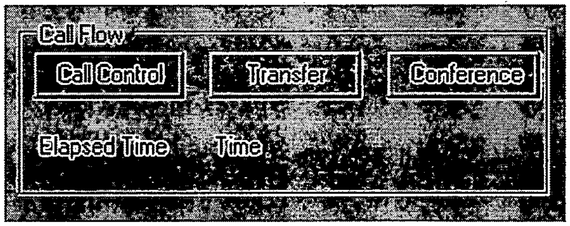
A smaller text area, where the agent can type the new message to the customer and press the send button.

Will have the capability for storing a pre defined set of messages, which the agent can choose and send, instead of typing them. When clicked once on the message, message will be copied to message 'Send' area, and the agent can edit the same and send to the customer. If double clicked on the message, the message will be send directly to the other party in the session.

Will have the capability to log the history of the interactions in to a database, on a pre defined interval or on demand. We shall implement on demand storage in version 1.0.

Media Control Area

Voice Calls



Will have the following buttons.

Call flow control button

This button will have several states.

Accept State

When the request from UQE for the new call is accepted, the text on the button will display 'Accept' and the background color will be set to AGENT_CALL_ACCEPT_BG. When the accept is pressed, the following actions will take place:

- A message is sent to UQE
- A message is sent to Voice Call manager
- A MSMQ message is sent to Database, to indicate the call acceptance.
- The time on the call control is started, to indicate the elapsed time. The Timer control, will have the formatting to show the time in Minutes and Seconds (nn Mins : xx Secs). The font of the Time is changed to follow the Average Call time parameter set for Agents, which has been accessed from the data services during log on.
- The call wrap up area is enabled
- In The extended presence, the channel presence start time is updated and the status is set to 'On Call'
- The call flow button status is changed to 'End Call', the background color is changed to AGENT_CALL_ENDCALL_BG

End Call state

In this state, the agent continues to talk with the customer. When this voice call is completed, and if required, keeps logging the information about the call in the wrap up area. When the voice call is completed, the agent presses this button.
When the agent presses this button the following activities will take place.

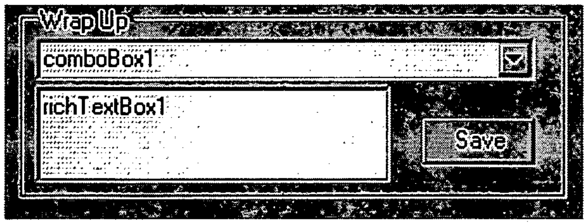
- The button state is changed to 'Ready'. The background color is changed to AGENT_CALL_READY_BG
- The timer value is stored to 'Time On Current Call", the timer value is initialized to collect wrap up time.
- A MSMQ message is sent to database
- End call message is sent to Call manager
- The status in the extended presence for the channel is updated
    - The channel presence status, for the instance is set to 'Wrap Up"
    - The number of calls is incremented by 1.
    - The total time is incremented by the current timer value.

Ready State

In the ready state, the agent will fill out any wrap up information, saves the information and then presses this button.
When this button is pressed, the following activities will take place.

- This button is disabled and the background is set to CCO_DEFAULT_BG
- The wrap up information is saved, if not saved earlier (sent as MSMQ Message).
- The timer value is taken in to current call wrap up time.
- The extended presence is updated.
    - The total wrap up time is incremented with the current timer value.
    - The channel entry in the extended presence is removed.
    - The number of active channels is reduced by 1.

Wrap up Area



This area will have the following fields :

- Combo box, populated with pre configured list of possible wrap up codes. These are populated in the database. These are loaded in to the desktop during the agent log in process.
- Text area – Where the agent can enter a free form text.
- Save button – When pressed, saves the entered data in to the database. When ever pressed, a MSMQ message will be generated and send to database persistence object. When pressed for the first time, a new entry will be created, where as the subsequent saves, will update the previously entered data for the call. The save

button will be enable only, if there is some unsaved entry in the area, to avoid multiple messages.

Services

Log In services

The functionality of this service is to accept the user id and password once and log in to all the systems. For version 1.0 We will support a single sign on with any of the applications, which have a web based login functionality available. We need to explore the possibilities of using single sign on services like Microsoft Passport or bundling a single sign on solution as part of solution suite. In version 1.0 the systems, we need to support are:

- RTC Server
- CCO App server through share point services
- Microsoft CRM Application

The log in services configurations are stored as part of the system. Some configurations are part of the client application. And some may be part of server application.

In addition log in may involve registering the agent with any other services like UQE and call manager etc.

Once the login is successful, we need to cache the session information, at the client, so that all the information we need for further interaction with the system are available locally.

Message interaction services

The functionality of this service is to provide interfaces for the other sub systems interact with desktop application. These services may be implemented with several transportation mechanism. Here are the once we will implement in Version 1.0

.NET Remote services

The Messaging services for the UQE and the call manager to interact with the agent desk top, will be implemented as .NET services with binary object as parameters.

newContactRequest

This message is sent by UQE, when ever a new call is allocated to the agent and the agent is available as per the presence info in the UQE's RTC Server.
On receiving this message, the following activities will take place on agent's desk top.

- Check if the overall status of the agent is available, if status is not available, then reject the message.
- Check whether the agent can handle this message based on the maximum instances the agent can handle for the given channel verses the active channels, If capacity is available, then increment the number of active instances for that channel. Return the accept token. If the capacity is not available return the reject token.
- Call the appropriate application area tab with all the relevant information
- Enable the media control to be in 'Accept' State.

Parameters: This service will accept the call object as the parameter, which may contain

- Call Number – A unique ID, either passed on from IVR or generated internally.
- Call Type – New, Transfer or conference
- Channel Type – Voice, IM, Email
- Enterprise ID
- CCO ID
- Customer ID
- Session ID
- IVR Channel ID

contactAcceptMessage

When the agent presses the Accept button on the media control area, send this message to the UQE using the .NET remote. Will pass the call object, which is associated with this call, as parameter.

Send the accept message to the call manager with call object as the parameter.

cancelContactRequest

This message is sent by UQE, when ever the previously routed call to the agent is cancelled due to various reasons. Upon receiving this message, if the call is not accepted by the agent yet, it will cancel the call referred in the message.

MSMQ Messages

The agent desk top will raise the MSMQ Message, for reporting purposes. The message format is as follows:

- MessageType – "Call"
- Call Number – A unique ID, either passed on from IVR or generated internally.
- Call Type – New, Transfer or conference
- Channel Type – Voice, IM, Email
- Enterprise ID
- CCO ID
- IVR Channel ID
- Agent ID
- Event Time
- Event Type – Depends on the time when it is raised.

The events when the application to raise the MSMQ messages are:

- The call received at Desk Top, event type will be AGENT_CALL_RECD or AGENT_CALL_RECD_REJECT
- The agent accepts the call, event type will be AGENT_CALL_ACCEPT
- The agent ends the call, event type will be AGENT_CALL_END
- The agent presses the ready button, event type will be AGENT_CALL_WRAPPED

In addition to the above the agent application will also raise messages, which will affect the agent status in RTC. These messages are also sent to MSMQ for persistence. The message format for this type of messages are:

- MessageType – "AgentStatus"
- PresenceStatus – One of the enumerated values for the presence.
- Agent ID
- Event Time
- Event Type – Depends on the time when it is raised.

Application audit / management services

Under these services we will implement the following capabilities.

Desk top Image logging

Log the screen shot of the agent at a pre defined events / time interval. The images will be sent as MSMQ messages for storing in to a centralized storage and retrieval services. We will also provide a button to perform this function, as part of the wrap up. In version 1.0 we will implement log on demand.

Application events log

In version 1.0 we will log all the application errors in to Application events log. Using the .NET application error log application block.
Later implement the log in to a central repository in Ver 2.0

Desk top sharing with the supervisor

The objective of this feature is to enable the agent desktop application to be monitored by the supervisor in real time. This will have the ability for the supervisor to see the Agents application interaction, with out the knowledge of the Agent. We will use the RTC or Net Meeting capability for implementing the same. In version 1.0 we will implement the explicit application sharing scenario, where by the supervisor initiates the request, and will be allowed to watch the agent desk top only after the agent accepts the request.

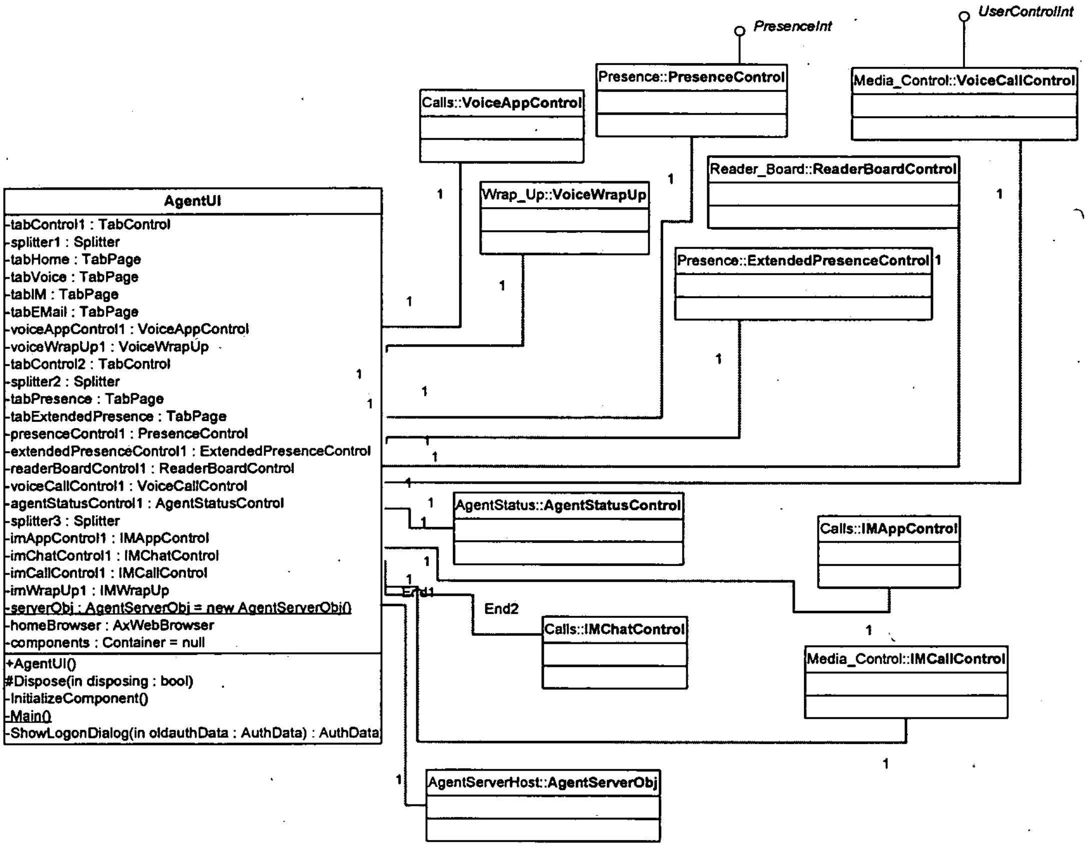
Interfaces

Agent Desktop Interfaces:

```
namespace AgentInterFaces
{
      /// <summary>
      /// Main Interface for clients to call.
      /// </summary>
      public interface IAgentServer
      {
            /// <summary>
            /// Resereves a channel for a call
            /// </summary>
            /// <returns>true if channel is reserved, otherwise
false</returns>
            bool ReserveCallRequest(NewCallRequest obj);

            /// <summary>
            /// Cancels an existing call and free that channel
            /// </summary>
            void CancelCallRequest(string CustomeRequestId);
      }

      /// <summary>
      /// Enumrates Call Type
      /// </summary>
      [Serializable]
      public enum CALL_TYPE
      {
            NEW,
            TRANSFER,
            CONFERENCE,
            CONSULTATION
      }

      /// <summary>
      /// Enumerates Channel Type
      /// </summary>
      [Serializable]
      public enum CHANNEL_TYPE
      {
            VOICE,
            IM,
            EMAIL
      }


      /// <summary>
      /// Encapsulates all the data for new call
      /// </summary>

      [Serializable]
```

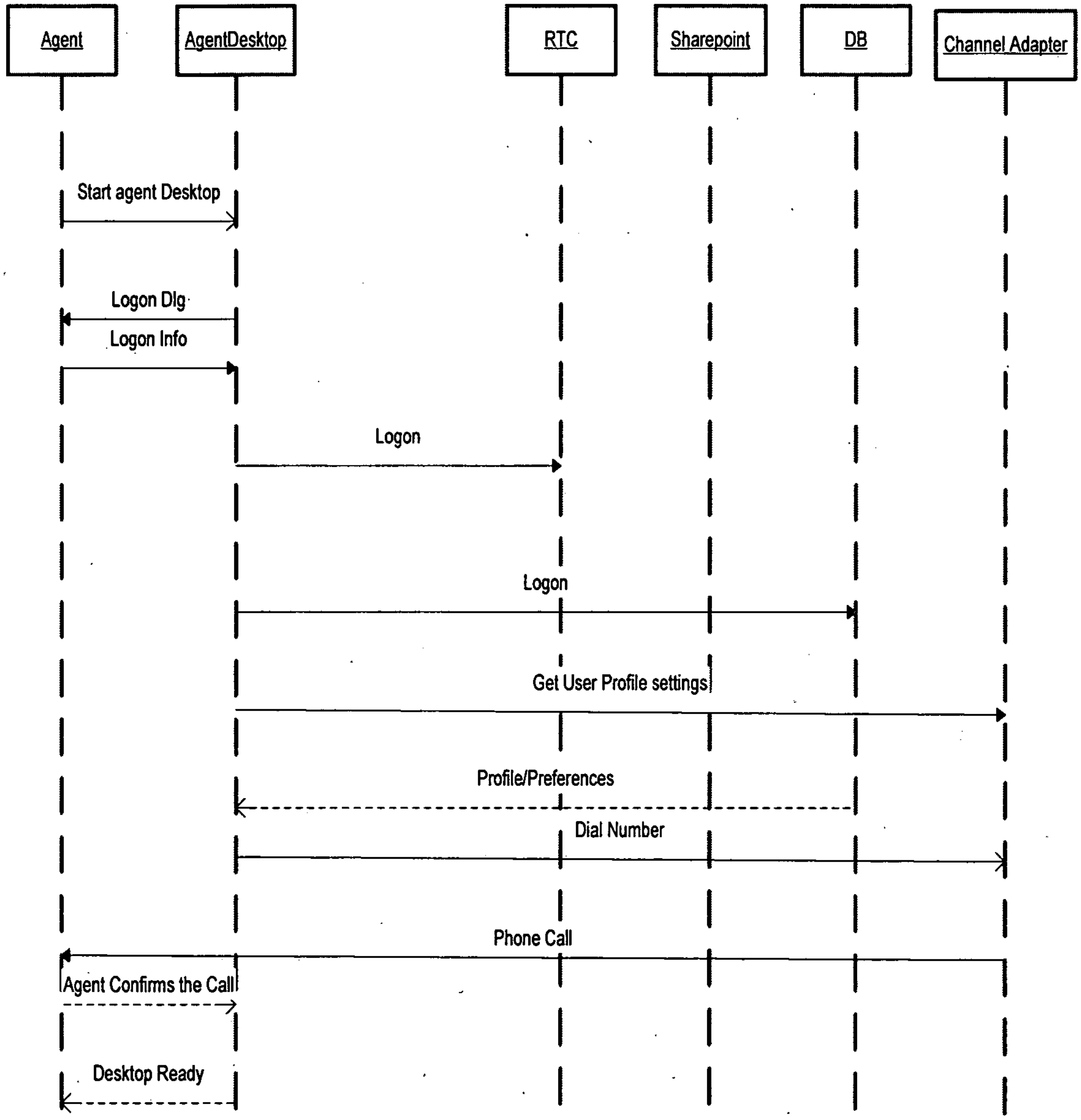
```
    public class NewCallRequest
    {
        /// <summary>
        /// Unique ID for the call
        /// </summary>
        public string CustomeRequestId;

        /// <summary>
        /// Call type
        /// </summary>
        /// <value>Value from CALL_TYPE</value>
        public CALL_TYPE CallType;

        /// <summary>
        /// Channel Type
        /// </summary>
        /// <value>Value from ChannelTypeEnum</value>
        public CHANNEL_TYPE ChannelType;

        /// <summary>
        /// ID of the entreprise
        /// </summary>
        public string EnterpriseId;

        /// <summary>
        /// Customer ID
        /// </summary>
        public string CustomerId;

        /// <summary>
        /// Initial Channle URI
        /// </summary>
        public string ChannelURI;
    }
}
```

Class Diagram for Agent Desktop

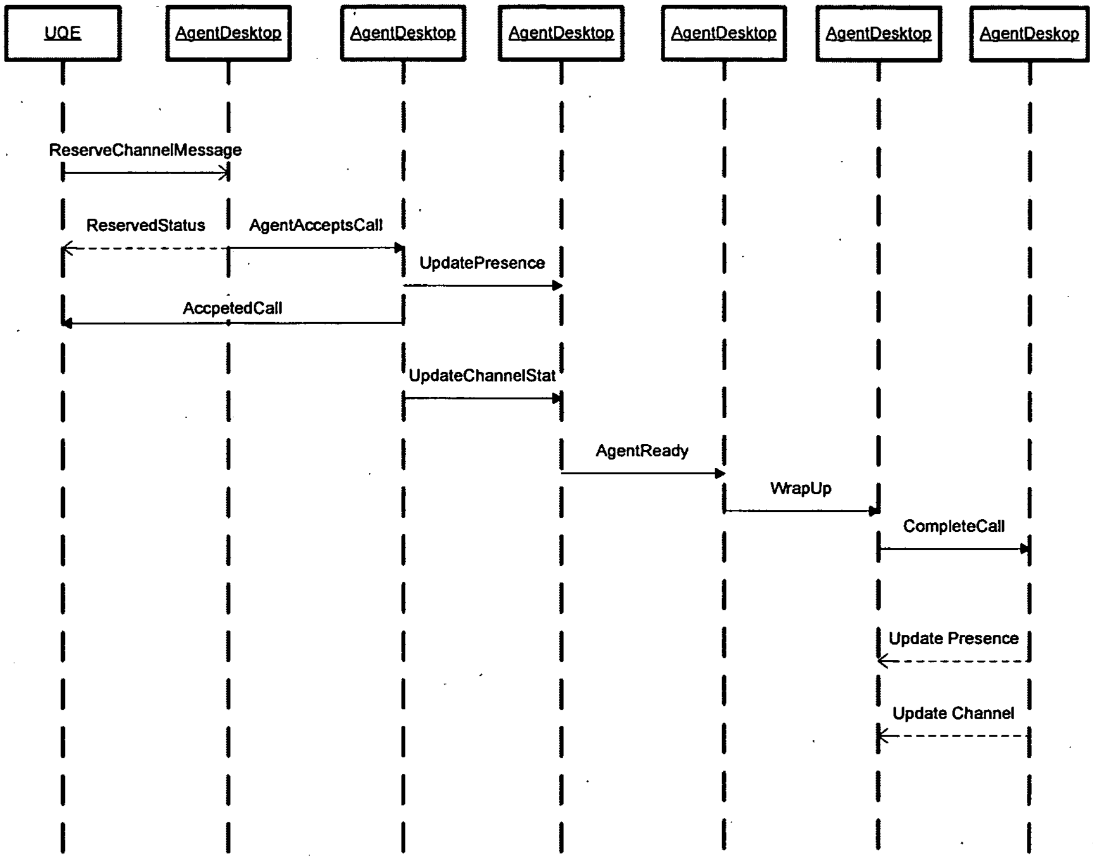

PresenceInt
UserControlInt
Presence::PresenceControl
Media_Control::VoiceCallControl
Calls::VoiceAppControl
Reader_Board::ReaderBoardControl
Wrap_Up::VoiceWrapUp
Presence::ExtendedPresenceControl
AgentUI
-tabControl1 : TabControl
-splitter1 : Splitter
-tabHome : TabPage
-tabVoice : TabPage
-tabIM : TabPage
-tabEMail : TabPage
-voiceAppControl1 : VoiceAppControl
-voiceWrapUp1 : VoiceWrapUp
-tabControl2 : TabControl
-splitter2 : Splitter
-tabPresence : TabPage
-tabExtendedPresence : TabPage
-presenceControl1 : PresenceControl
-extendedPresenceControl1 : ExtendedPresenceControl
-readerBoardControl1 : ReaderBoardControl
-voiceCallControl1 : VoiceCallControl
-agentStatusControl1 : AgentStatusControl
-splitter3 : Splitter
-imAppControl1 : IMAppControl
-imChatControl1 : IMChatControl
-imCallControl1 : IMCallControl
-imWrapUp1 : IMWrapUp
-serverObj : AgentServerObj = new AgentServerObj()
-homeBrowser : AxWebBrowser
-components : Container = null
+AgentUI()
#Dispose(in disposing : bool)
-InitializeComponent()
-Main()
-ShowLogonDialog(in oldauthData : AuthData) : AuthData
AgentStatus::AgentStatusControl
Calls::IMAppControl
Calls::IMChatControl
Media_Control::IMCallControl
AgentServerHost::AgentServerObj
End1
End2
1

Sequence Diagrams

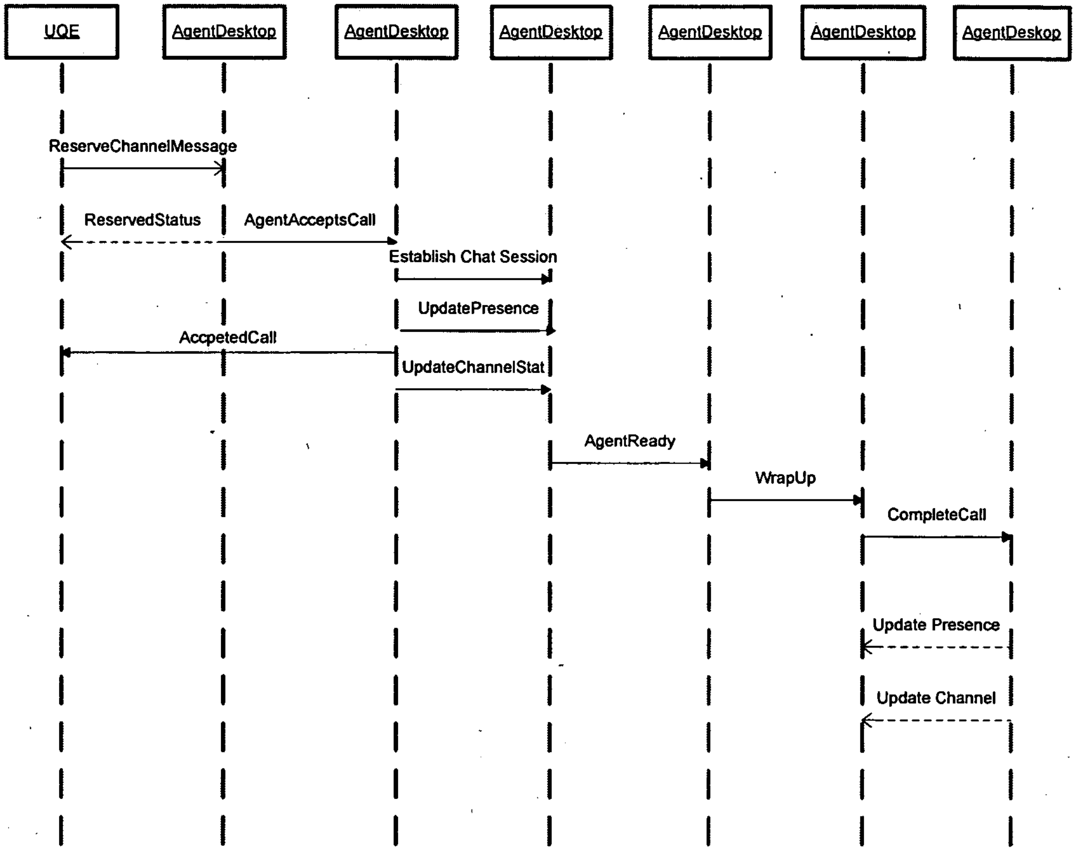
Login Sequence



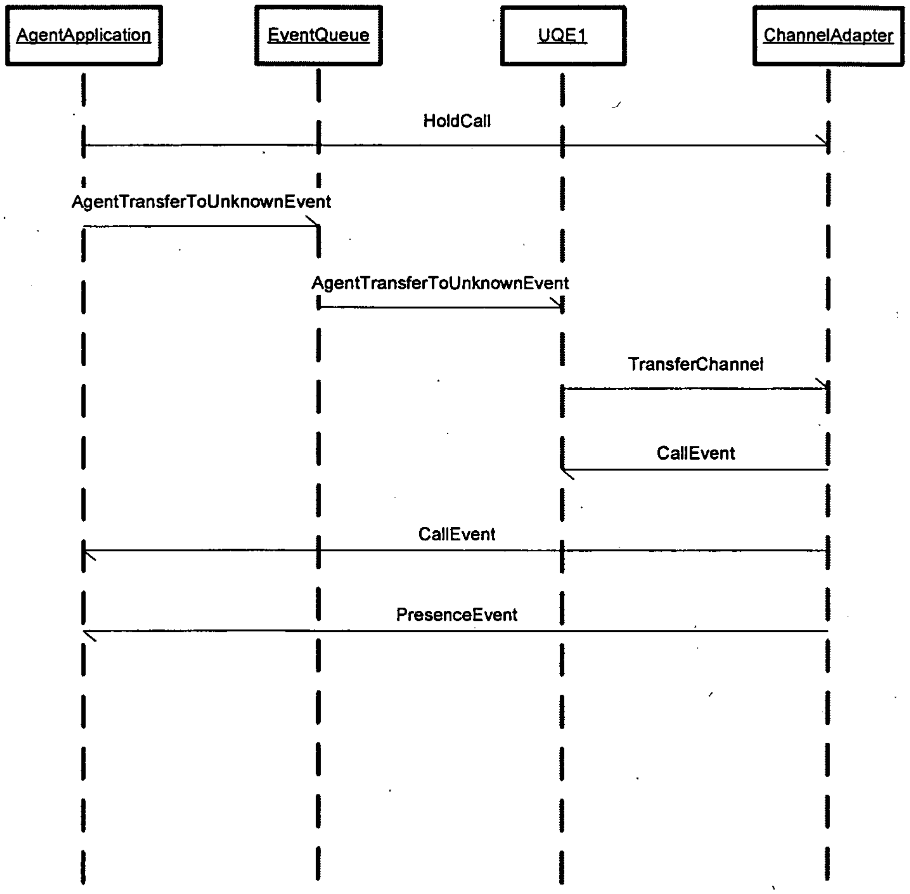
Voice call Sequence



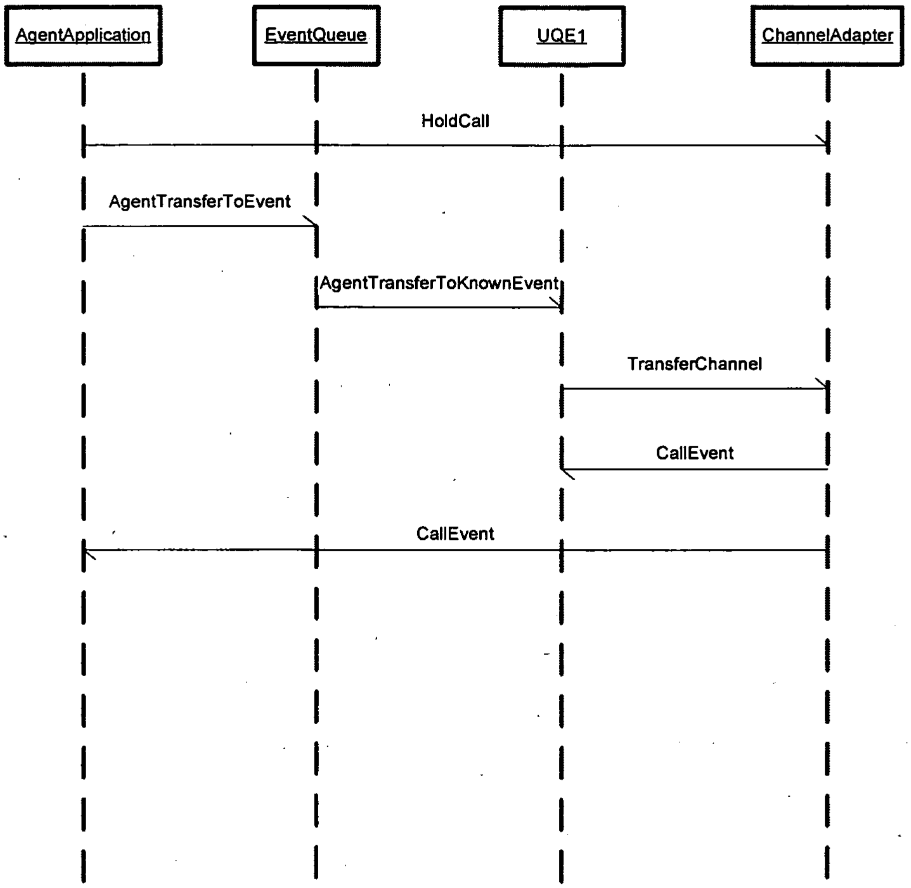
IM call Sequence



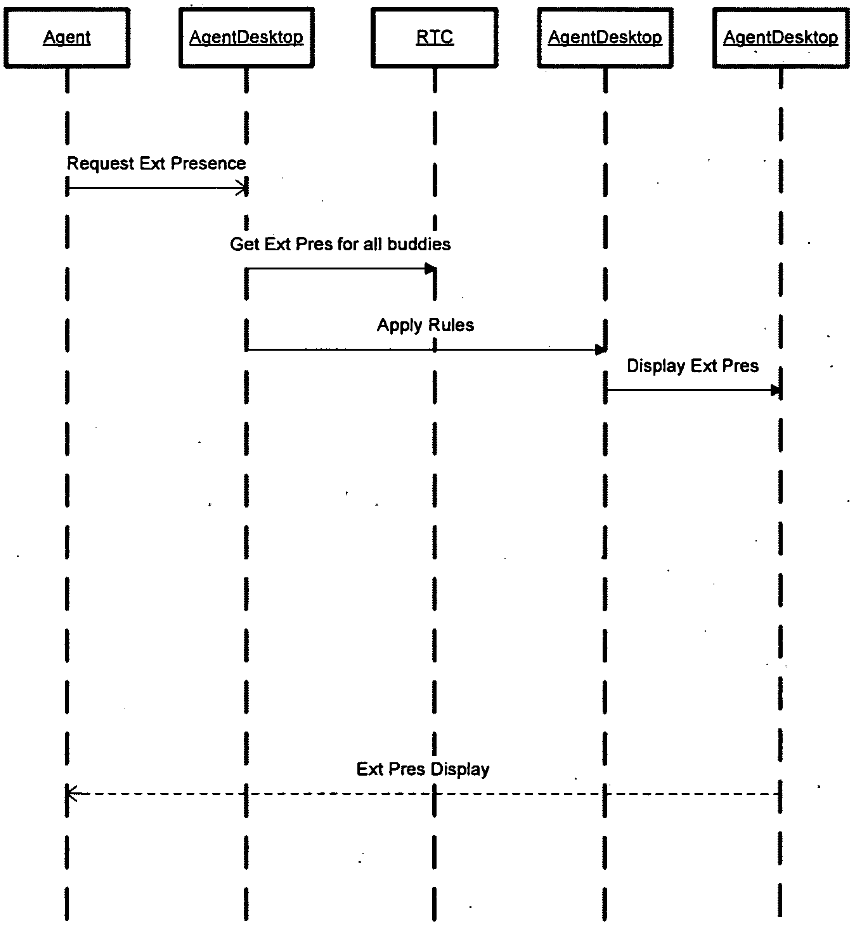
Call Transfer to unknown



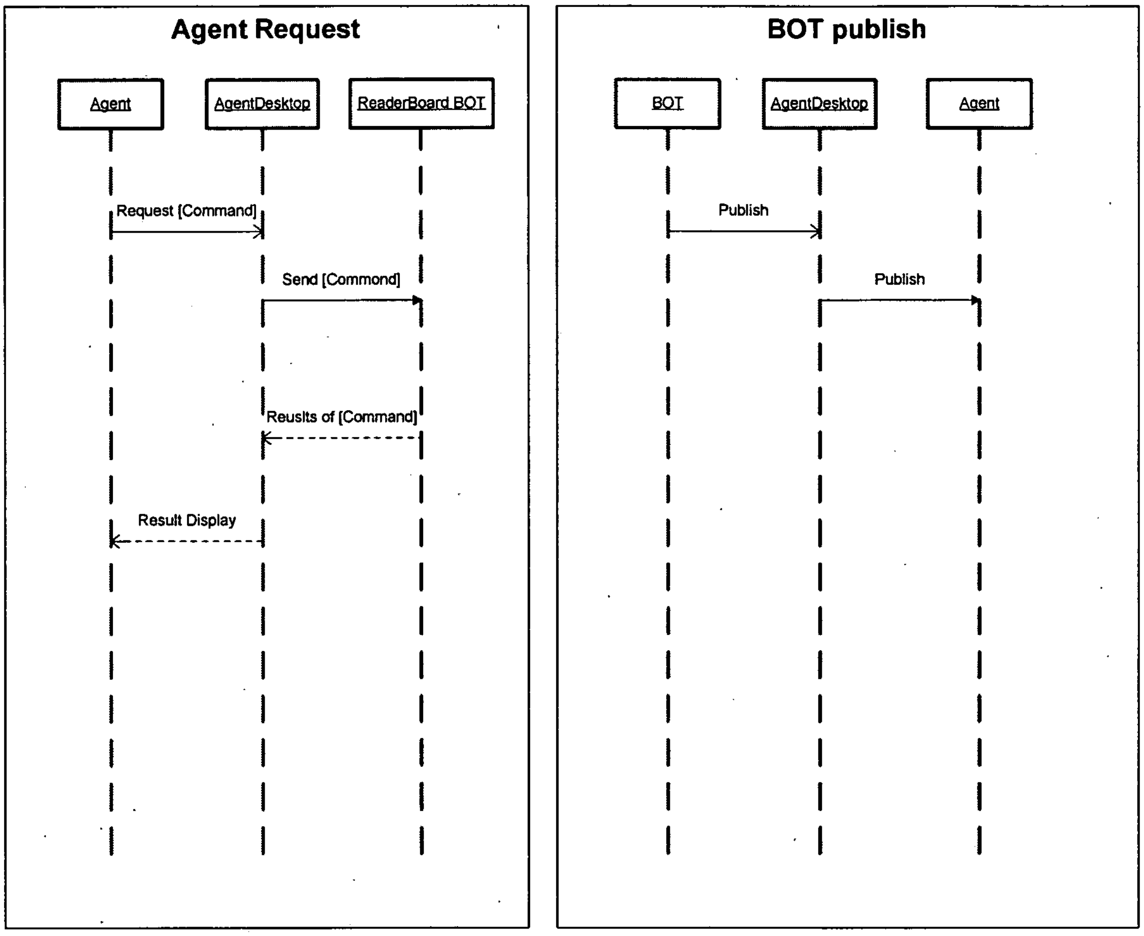
Call Transfer to Known



Extended Presence



Reader Board


Agent Request
Agent
AgentDesktop
ReaderBoard BOT
Request [Command]
Send [Commond]
Reuslts of [Command]
Result Display
BOT publish
BOT
AgentDesktop
Agent
Publish
Publish